

A Possibilistic Valid-time Model

José Enrique Pons¹, Christophe Billiet², Olga Pons Capote¹, and Guy De Tré²

¹ Department of Computer Science and Artificial Intelligence, Universidad de Granada, Escuela Técnica Superior de Ingeniería Informática
C/Periodista Daniel Saucedo Aranda s/n, E-18071, Granada, Spain
`jpons,opc@decsai.ugr.es`

² Department of Telecommunications and Information Processing, Ghent University
Sint-Pietersnieuwstraat 41 B-9000, Ghent, Belgium.
`Christophe.Billiet,Guy.De.Tre@ugent.be`

Abstract. Information in databases can be imperfect and this imperfection has several forms and causes. In some cases, a single value should be stored, but it is (partially) unknown. The uncertainty about which value to store leads to the aforementioned imperfection. In temporal databases, uncertainty can arise, concerning which temporal notion needs to be stored. Because in temporal databases, temporal notions influence the consistency with which the database models the reality, this uncertainty has a direct impact on the consistency of the model. To represent this temporal uncertainty, previous works have adapted fuzzy sets with conjunctive interpretation, an approach that might prove misleading. This work presents a model that represents the uncertainty using possibility and necessity measures, which are fuzzy sets with disjunctive interpretations.

Keywords: temporal databases, fuzzy databases, information systems, incompleteness

1 Introduction

Dealing with time in information systems is usually a difficult task [4]. First of all, the concept of *time* itself is very complex. More towards information systems, several proposals have been concerned with obtaining theoretical models that allow the representation of time in information systems [6]. Notably, temporal relations between time intervals and to a lesser extent time points, were studied by Allen [1].

One issue in these attempts at temporal modelling is of course the inherent complexities of the concept of time. Another is the possible inherent imperfections in the description of notions of time. The description of the temporal notion in a sentence like ‘The Alhambra, which I can see from my room, is a few hundred years old.’ contains imperfection caused by the vagueness of the used expression. The description of the temporal notion in a sentence like ‘The Alhambra was finished somewhere between 1301 A.D. and 1400 A.D.’ contains

imperfection caused by the imprecision in the used expression about the exact finishing day of the building. Though precise boundaries are given, the exact day on which the building was finished is unknown. An example of uncertainty is given by the following sentence: ‘I believe that the Alhambra was finished in 1300 A.D.’

The transition between temporal granularities [17] is also considered as a source of imprecision in temporal modelling and some proposals consider granularity as the base of the temporal model [6].

To allow information systems to cope with these imperfections, there are some approaches that work with probability [7], [21] whereas some other approaches adopt fuzzy sets for the representation of temporal information [2],[8]. The temporal relations studied by Allen were recently fuzzified by several authors [20],[23].

Next to the issues in temporal modelling, the inclusion of time in a relational database system leads to several more practical problems, as the modelling of the temporal aspects of a time-variant object has an impact on the database consistency. Different models have been proposed [15],[24],[19] that consider time as a crisp notion. Some of these models uses TSQL [16] (a temporal extension to SQL) as query language. In some specific works, authors consider the necessity of allowing imprecision in temporal models [6],[14].

Most of these models use the concepts of fuzzy numbers and fuzzy intervals to represent imprecise temporal notions and considerable attention has thus been given to the problem of transforming two fuzzy numbers which represent the boundaries of an imprecise time interval into one fuzzy interval representing this interval. There are several proposals for this transformation, but some, like those presented in [14] are misleading.

In this work we will propose a possibilistic valid-time model for relational databases, based on the framework presented in [22], which deals with the inference of uncertainty about the evaluation of sets from the uncertainty about the exact values used in evaluation.

The remainder of the paper is organized as follows. In section 2, some preliminaries and concepts are presented. Section 3 presents the possibilistic model for valid-time databases. In this section, both representation and querying are introduced. Finally, section 4 presents the main conclusions and some lines for future work in this research.

2 Preliminaries

In this section, some basic concepts are introduced concerning possibilistic variables and fuzzy numbers and intervals, after which the framework of set evaluation by ill-known constraints [22] is explained. The section concludes with a brief introduction to temporal databases.

2.1 Possibilistic Variables

Possibilistic variables rely on possibility theory [10]. A *possibilistic variable* is defined as follows [22].

Definition 1. A *possibilistic variable* X over a universe U is defined as a variable taking exactly one value in U , but for which this value is (partially) unknown. Its possibility distribution π_X on U models the available knowledge about the value that X takes: for each $u \in U$, $\pi_X(u)$ represents the possibility that X takes the value u . In this work, this possibility is interpreted as a measure of how plausible it is that X takes the value u , given (partial) knowledge about the value X takes.

The exact value a possibilistic variable takes, which is (partially) unknown, is called an *ill-known value* in this work [10].

When a possibilistic variable is defined on the powerset $\mathcal{P}(R)$ of some universe R , the unique value the variable takes will be a crisp set and its possibility distribution on the powerset $\mathcal{P}(R)$ will describe the possibility of each crisp subset of R to be the value the variable takes. This exact value (a crisp set) the variable takes, is now called an *ill-known set* [10].

A specific application of possibilistic variables is obtained when the universe under consideration is the set of Boolean values $\mathbb{B} = \{T, F\}$. Indeed, any Boolean proposition p takes just one value in \mathbb{B} . If the knowledge about which value this proposition p will take is given by a possibility distribution π_p , the proposition can be seen as a possibilistic variable. As the interest lies with the case where the proposition holds, the possibility and necessity that $p = T$ (the proposition holds) demand most attention. This possibility and necessity is noted here as:

$$\text{Possibility that } p = T \text{ (p holds):} \quad Pos(p) = \pi_p(T) \quad (1)$$

$$\text{Necessity that } p = T \text{ (p holds):} \quad Nec(p) = 1 - \pi_p(F) \quad (2)$$

This work will deal with ill-known intervals. These are ill-known sets, defined and represented via a start and end point, which will be ill-known values. The elements of the set are the values between the start and end point. A closed ill-known interval with start point defined by possibilistic variable X and end point by possibilistic variable Y is noted here $[X, Y]$. The correspondences and transitions between the representations of ill-known sets, between the representations of ill-known intervals and between the representations of an ill-known set and an ill-known interval are part of the authors current research.

2.2 Fuzzy Numbers and Fuzzy Intervals

Among others, Dubois and Prade [9] use fuzzy sets [25] to define a *fuzzy interval*:

Definition 2. A *fuzzy interval* is a fuzzy set M , defined by a membership function μ_M , on the set of real numbers \mathbb{R} such that:

$$\mu_M : \mathbb{R} \rightarrow [0, 1]$$

$$\forall(u, v) \in \mathbb{R}^2 : \forall w \in [u, v] : \mu_M(w) \geq \min(\mu_M(u), \mu_M(v)) \quad (3)$$

$$\exists m \in \mathbb{R} : \mu_M(m) = 1 \quad (4)$$

If this modal value m is unique, then M is referred to as a *fuzzy number*. In other words, if the core of a fuzzy interval is a singleton, it is referred to as a fuzzy number.

The most convenient form of the membership function of a fuzzy number is a triangular form. It can be shown that such a membership function μ_M for a fuzzy number M is convex and normalized. Three real values, denoted by a , b and D , suffice to represent a triangular membership function of a fuzzy number and in this work, a fuzzy number defined as such will be noted as $[D, a, b]$. Here:

- D denotes the single value in the core of M
- $D - a$ is then $\inf\{u \in \mathbb{R} : \mu_M(u) > 0\}$
- $D + b$ is then $\sup\{u \in \mathbb{R} : \mu_M(u) > 0\}$

2.3 Interval Evaluation by Ill-known Constraints

The problem of interval evaluation is more generally explained in [22]: the need exists to know if all points in a crisp interval I reside between the boundaries of an ill-known interval $[X, Y]$. In [22], the notion of an *ill-known constraint* is introduced:

Definition 3. *Given a universe U , an ill-known constraint C on a set $A \subseteq U$ is specified by means of a binary relation $R \subseteq U^2$ and a fixed, ill-known value denoted by its possibilistic variable V over U , i.e.:*

$$C \triangleq (R, V) \quad (5)$$

Set A now satisfies the constraint if and only if:

$$\forall a \in A : (a, V) \in R \quad (6)$$

An example of an ill-known constraint is $C_{ex} \triangleq (<, X)$. Some set A then satisfies C_{ex} if $\forall a \in A : a < X$, given possibilistic variable X .

The satisfaction of a constraint $C \triangleq (R, V)$ by a set A is basically still a Boolean matter, but due to the uncertainty about the ill-known value V , it can be uncertain whether C is satisfied by A or not [22]. In fact, this satisfaction now behaves as a proposition. Based on the possibility distribution π_V of V , the possibility and necessity that A satisfies C can be found. This proposition can thus be seen as a possibilistic variable on \mathbb{B} . The required possibility and necessity are:

$$\text{Pos}(A \text{ satisfies } C) = \min_{a \in A} \left(\sup_{(a, w) \in R} \pi_V(w) \right) \quad (7)$$

$$\text{Nec}(A \text{ satisfies } C) = \min_{a \in A} \left(\inf_{(a, w) \notin R} 1 - \pi_V(w) \right) \quad (8)$$

Now, e.g., to check if crisp interval $I = [j, k]$ is included in $[X, Y]$, 2 ill-known constraints are constructed:

$$C_1 \triangleq (\geq, X) \quad (9)$$

$$C_2 \triangleq (\leq, Y) \quad (10)$$

To calculate the possibility and necessity concerning a conjunction of constraints, the min operator can be used. The possibility and necessity of I being included in $[X, Y]$ are now:

$$\text{Pos}(I \text{ satisfies } C_1 \text{ and } C_2) = \min_{a \in I} \left(\sup_{a \geq w} \pi_X(w), \sup_{a \leq v} \pi_Y(v) \right) \quad (11)$$

$$\text{Nec}(I \text{ satisfies } C_1 \text{ and } C_2) = \min_{a \in I} \left(\inf_{a < w} 1 - \pi_X(w), \inf_{a > v} 1 - \pi_Y(v) \right). \quad (12)$$

2.4 Temporal Databases

A *temporal database* is a database that manages some aspects of time in its schema [12]. The reality a temporal database tries to model, contains some temporal notions which have to be handled specifically in order to maintain a consistent modelling behavior. A *chronon* is the shortest duration of time supported by the database. Time can be represented either as points [11] or intervals [14],[3] that may be subject to imperfection.

The temporal notions in temporal databases can be classified into four types based on their interpretation and modelling purpose. *User-defined time* has no specific impact on the database consistency, but the other types do:

- *transaction time* [15]: time when the fact is stored in the database.
- *valid time* [24]: time when the fact is true in the modelled reality.
- *decision time* [19]: time when an event was decided to happen.

Database models can also be classified into *bi-temporal* (both valid and transaction-time) or *tri-temporal* (bi-temporal and decision time) models.

3 Proposal

In this section, a valid-time model, able to handle imperfect valid-time notions, will be presented. First, the representation and storage of valid-time will be presented. Next, an approach to query the model is proposed. Finally, an example is given.

3.1 Valid-time Intervals

Representation A database models real objects by storing the values of an object for each attribute describing a property of the object. Thus, a valid-time database following the presented proposal will model the time during which an object in a certain state is valid, by associating a *Possibilistic Valid-time Period* (PVP) to the record describing this object state:

Definition 4. A Possibilistic Valid-time Period is an ill-known interval in time, which models a time period during which an object in a certain state is valid.

Because a PVP is an ill-known interval, it allows modelling the uncertainty about the start and/or end point of a time interval (and thus about the time interval itself) if such uncertainty exists. The interpretation is *disjunctive*: the PVP represents exactly one valid-time interval, but precisely *which* interval is represented, is (partially) unknown. In the presented model, only PVPs are considered of which the possibility distributions of the possibilistic variables defining the start and end point of the ill-known interval have exactly the same characteristics as the membership functions of fuzzy numbers. A perfectly known start or end point can then be modelled by such an ill-known value defined by a possibilistic variable P for which $\exists!x : \mu_P(x) > 0$.

As mentioned in [22], this approach differs from the one where a valid-time period is represented by one fuzzy set. Such a fuzzy set is seen as a possibility distribution on \mathbb{R} and thus defines just one ill-known value. However, in the presented approach, a time period is modelled using an ill-known set, which is defined by a possibility distribution on $\mathcal{P}(\mathbb{R})$.

Storage To store a PVP, the possibility distributions defining the ill-known start and end point are stored. To store such a possibility distribution, the representation as used in the fuzzy interface for relational databases *FIRST* [18], [13] is used. Using this representation, it is possible to represent not only fuzzy numbers, but also (fuzzy) constants. To store an ill-known value, four values (FT, F1, F2 and F3) are stored. They are explained in Table 1. Note that while NULL denotes the fuzzy constant NULL, N denotes a null value in the database [18].

Table 1. Relational representation for an ill-known time point. FT denotes Fuzzy Type. Field FT indicates that the values stored in F1, F2 and F3 denote either NULL, UNKNOWN, UNDEFINED or a triangular possibility distribution $M = [D, a, b]$. Fields F1, F2 and F3 contain the actual values.

Value	FT	F1	F2	F3	$\mu(x)$	Description
UNKNOWN	0	N	N	N	1	Any value is equally possible
UNDEFINED	1	N	N	N	0	The value is not defined
NULL	2	N	N	N	not defined	Nothing is known about the value
M	3	D	a	b	μ_M	Ill-known value

3.2 Querying Ill-known Valid-time Intervals

This subsection discusses a tool for querying. The focus will evidently lie on the querying of valid-time periods. First, the query structure will be defined, followed by the evaluation and ranking methods.

Query Structure It is important to notice that every valid-time notion in the database is represented using a PVP. In the presented framework, a query has two separate parts:

Definition 5. A query \tilde{Q} is specified by:

$$\tilde{Q} = (Q^{time}, Q) \quad (13)$$

Here, Q denotes the collection of (possibly fuzzy) non-temporal preferences of the user and Q^{time} denotes the temporal constraint given by the user:

Definition 6. Q^{time} is defined by:

$$Q^{time} = (I, AR) \quad (14)$$

Here, I is a crisp time interval and AR is one of Allen's relations [1]. The interpretation is that for a record with valid-time period given by a PVP J , the user requires that $I \text{ } AR \text{ } J$ holds.

Query Evaluation In fuzzy querying of regular (relational) databases, the modelling of query satisfaction is a matter of degree. Usually, the evaluation of the query requirements for a record results in a satisfaction degree s , where s lies in $[0, 1]$, where 0 denotes total dissatisfaction and 1 denotes complete satisfaction. In crisp querying, the evaluation of query requirements for a record results in the accepting or rejecting of the record as a part of the result set. This can be modelled using satisfaction degrees, by assigning rejection a degree of 0 and acceptance a degree of 1 and not using any other value in $[0, 1]$.

The evaluation of a query $\tilde{Q} = (Q^{time}, Q)$, where $Q^{time} = (I, AR)$ is now handled as follows. For each record r in the database, with the valid-time notion of r being specified by a PVP J , two things happen independently:

- The preferences expressed in Q are evaluated, resulting in a satisfaction degree denoted here $e_Q(r)$. The presented model accepts any sound way of calculating this evaluation, as long as $e_Q(r) \in [0, 1]$.
- Depending on AR , a specific set of ill-known constraints is considered, which can be found in Table 2. The possibility and necessity that r fulfills all these constraints are calculated using formulas based on equations (7) respectively (8) and aggregated using the min operator.

Ranking and Aggregation In order to present the results to the user, a crude ranking method is used: for every record r , the sum of $\text{Pos}_{Q^{time}}(r)$ and $\text{Nec}_{Q^{time}}(r)$ gives an evaluation score $e'_{Q^{time}}(r)$ in interval $[0, 2]$. Because necessity cannot exceed 0 unless possibility is 1, this gives a natural ranking score. Some authors [5] mentioned before that the possibility and necessity measures result in a total order in the set of events. This $e'_{Q^{time}}(r)$ is then rescaled to the unit interval, resulting in $e_{Q^{time}}(r)$. The final ranking $e_{final}(r)$ is now given by a convex combination:

Table 2. Allen’s relations used in the framework. Here, $I = [a, b]$ denotes the PVP in the query, $J = [X, Y]$ denotes the PVP of the record, with π_X and π_Y the possibility distributions of X and Y respectively. For each of Allen’s relations AR , the corresponding value in column ‘Constraints’ gives the constraints corresponding to AR . The last column contains the corresponding formula to calculate the possibility that I satisfies all constraints given in column ‘Constraints’, which are based on equation (7).

Allen Relation	Constraints	Pos(I satisfies all constraints $C_i, i = 1, 2, \dots$)
I before J	$C_1 \triangleq (<, X)$	$\sup_{a > w} \pi_x(w)$
I equal J	$C_1 \triangleq (\geq, X), C_2 \triangleq (\neq, X)$ $C_3 \triangleq (\leq, Y), C_4 \triangleq (\neq, Y)$	$\min(\sup_{a \leq w} \pi_x(w), \pi_x(w),$ $\sup_{b > w} \pi_Y(w), \pi_Y(w))$
I meets J	$C_1 \triangleq (\leq, X), C_2 \triangleq (\neq, X)$	$\min(\sup_{a > w} \pi_X(w), \pi_X(w))$
I overlaps J	$C_1 \triangleq (<, Y), C_2 \triangleq (\leq, X)$ $C_3 \triangleq (\geq, X)$	$\min(\sup_{b > w} \pi_Y(w),$ $\sup_{a > w} \pi_X(w), \sup_{a < w} \pi_X(w))$
I during J	$C_1 \triangleq (>, X), C_2 \triangleq (\leq, Y)$ $C_3 \triangleq (\geq, X), C_4 \triangleq (<, Y)$	$\max(\min(\sup_{a < w} \pi_X(w), \sup_{b \geq w} \pi_Y(w)),$ $\min(\sup_{a \leq w} \pi_X(w), \sup_{b > w} \pi_Y(w))$
I starts J	$C_1 \triangleq (\geq, X), C_2 \triangleq (\neq, X)$	$\min(\sup_{a \leq w} \pi_X(w), \pi_X(w))$
I finishes J	$C_1 \triangleq (\leq, Y), C_2 \triangleq (\neq, Y)$	$\min(\sup_{b > w} \pi_Y(w), \pi_Y(w))$

$$e_{final}(r) = \omega * e_Q(r) + (1 - \omega) * e_{Qtime}(r), \omega \in [0, 1] \quad (15)$$

The use of this convex combination allows a record to make up for a low score for the temporal constraint by a good score for the non-temporal constraint (or vice versa). Changing ω also allows granting the temporal constraint more weight with respect to the non-temporal constraint (or vice versa).

Example 1. Consider the example relation given in Table 3 describing car models, containing general attributes (model name, manufacturer, car segment) and one temporal attribute (stored as explained in subsection 3.1) describing the approximate period of time during which the car model was sold. The value for D is stored in *yyyy* format and a and b are represented by an integer. The ID field identifies a car model while the field Instance ID (IID) identifies the instance for a car model, thus a car model in a certain state.

Table 3. Example database

ID	IID	Segment	Manufacturer	Name	Start	End
001	1	B	Peugeot	205	[1985,2,3]	[1997,2,1]
002	1	C	Peugeot	305	[1977,2,2]	[1989,2,3]
003	1	B	Citroen	C2	[2002,2,2]	[2006,1,1]
001	2	B	Peugeot	206	[2000,1,2]	[2011,2,1]
001	3	B	Peugeot	207	[2006,1,1]	[2011,1,1]

Consider the following query:

The user wants to obtain a list of models from segment B, made by manufacturer Peugeot before the year interval 2001-2005

Using the introduced notations in (13), the query is translated to:

$$Q^{time} = ([2001, 2005], before) \quad (16)$$

$$Q = (Segment = B) \wedge (Manufacturer = Peugeot) \quad (17)$$

Table 4. Result table and ranking

ID	IID	Pos $_{Q^{time}}$	Nec $_{Q^{time}}$	$e_{Q^{time}}(rescaled)$	Q	$e_{final}(\omega = 0.5)$
001	1	1	1	1	1	1
002	1	1	1	1	0.5	0.75
003	1	1	0.5	0.75	0	0.375
001	2	1	0	0.5	1	0.75
001	3	0	0	0	1	0.5

Table 4 shows a natural and gradual ranking for the results. The last record, (ID 001, IID 3) shows also that with $\omega = 0.5$ both temporal and regular criteria have the same importance.

4 Conclusions and Further Research

We presented a valid-time model to represent and query ill-known temporal intervals. The framework tries to correctly represent and handle querying of temporal uncertainty in a database. It is possible to model the Allen's relations and also more complex relations. As future work, new ways for aggregation and ranking are considered.

Acknowledgements

Part of the researchers are supported by the grant BES-2009-013805 in the project TIN2008-02066: *Fuzzy Temporal Information treatment in RDBMS*.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM **26**, 832–843 (1983)
2. Billiet, C., Pons, J.E., Matthé, T., De Tré, G., Pons Capote, O.: Bipolar fuzzy querying of temporal databases. In: LNAI, vol. 7022, pp. 60–71. Springer, Ghent, Belgium (2011)
3. Bohlen, M., Busatto, R., Jensen, C.: Point-versus interval-based temporal data models. In: Data Engineering, 1998. Proceedings., 14th Int. Conf. on, pp. 192–200 (1998)

4. Bolour, A., Anderson, T.L., Dekeyser, L.J., Wong, H.K.T.: The role of time in information processing: a survey. *ACM SIGMOD Record* **12**, 27–50 (1982)
5. Bosc, P., Pivert, O.: Modeling and querying uncertain relational databases: A survey of approaches based on the possible worlds semantics. *Int. J. Uncertainty Fuzziness Knowledge-Based Syst.* **18**(5), 565–603 (2010)
6. Van der Cruyssen, B., De Caluwe, R., De Tré, G.: A theoretical fuzzy time model based on granularities. In: *Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing*, pp. 1127–1131. ELITE Foundation (1997)
7. Dekhtyar, A., Ross, R., Subrahmanian, V.S.: Probabilistic temporal databases, i: algebra. *ACM Trans. Database Syst.* **26**, 41–95 (2001)
8. Dubois, D., HadjAli, A., Prade, H.: Fuzziness and uncertainty in temporal reasoning. *Journal of Universal Computer Science* **9**(9), 1168–1194 (2003)
9. Dubois, D., Prade, H.: Ranking fuzzy numbers in the setting of possibility theory. *Information Sciences* **30**, 183–224 (1983)
10. Dubois, D., Prade, H.: *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York (1988)
11. DuBois, D., Prade, H.: Processing fuzzy temporal knowledge. *IEEE Transactions on Systems, Man, and Cybernetics* **19**, 729–744 (1989)
12. Dyreson, C., Grandi, F.e.a.: A consensus glossary of temporal database concepts. *SIGMOD Rec.* **23**, 52–64 (1994)
13. Galindo, J., Medina, J., Pons, O., Cubero, J.: A server for fuzzy sql queries. In: T. Andreasen, H. Christiansen, H. Larsen (eds.) *Flexible Query Answering Systems, LNCS*, vol. 1495, pp. 164–174. Springer Berlin / Heidelberg (1998)
14. Garrido, C., Marin, N., Pons, O.: Fuzzy intervals to represent fuzzy valid time in a temporal relational database. *Int. J. Uncertainty Fuzziness Knowledge-Based Syst.* **17**, 173–192 (2009)
15. Jensen, C.S., Mark, L., Roussopoulos, N.: Incremental implementation model for relational databases with transaction time. *IEEE Trans. Knowl. Data Eng.* **3**, 461–473 (1991)
16. Jensen, C.S., Snodgrass, R.T., Soo, M.D.: The tsql2 data model. In: *The TSQL2 Temporal Query Language*, pp. 153–238 (1995)
17. Lin, H., (codirector), C.J., Bohlen, M., Busatto, R., Gregersen, H., Torp, K., (codirector), R.S., Datta, A., Ram, S.: Efficient conversion between temporal granularities. Master's thesis, The University of Arizona (1997)
18. Medina, J., Pons, O., Cubero, J.: Gefred. a generalized model of fuzzy relational databases. *Information Sciences* **76**, 87–109 (1994)
19. Nascimento, M.A., Eich, M.H.: Decision time in temporal databases. In: *Proceedings of the 2nd Int. Workshop on Temporal Representation and Reasoning*, pp. 157–162 (1995)
20. Ohlbach, H.J.: Relations between fuzzy time intervals. *International Symposium on Temporal Representation and Reasoning* **0**, 44–51 (2004)
21. Parisi, F., Parker, A., Grant, J., Subrahmanian, V.S.: Scaling cautious selection in spatial probabilistic temporal databases. In: *Methods for Handling Imperfect Spatial Information*, pp. 307–340. Springer (2010)
22. Pons, J.E., Bronselaer, A., De Tré, G., Pons, O.: Possibilistic evaluation of sets (2011). Submitted to the *Int. J. Uncertainty Fuzziness Knowledge-Based Syst.*
23. Schockaert, S., De Cock, M., Kerre, E.: Fuzzifying allen's temporal interval relations. *Fuzzy Systems, IEEE Transactions on* **16**(2), 517–533 (2008)
24. Snodgrass, R.: The temporal query language tquel. In: *Proceedings of the 3rd ACM SIGACT-SIGMOD*, pp. 204–213. ACM, New York, NY, USA (1984)
25. Zadeh, L.: Fuzzy sets. *Information and Control* **8**, 338–353 (1965)